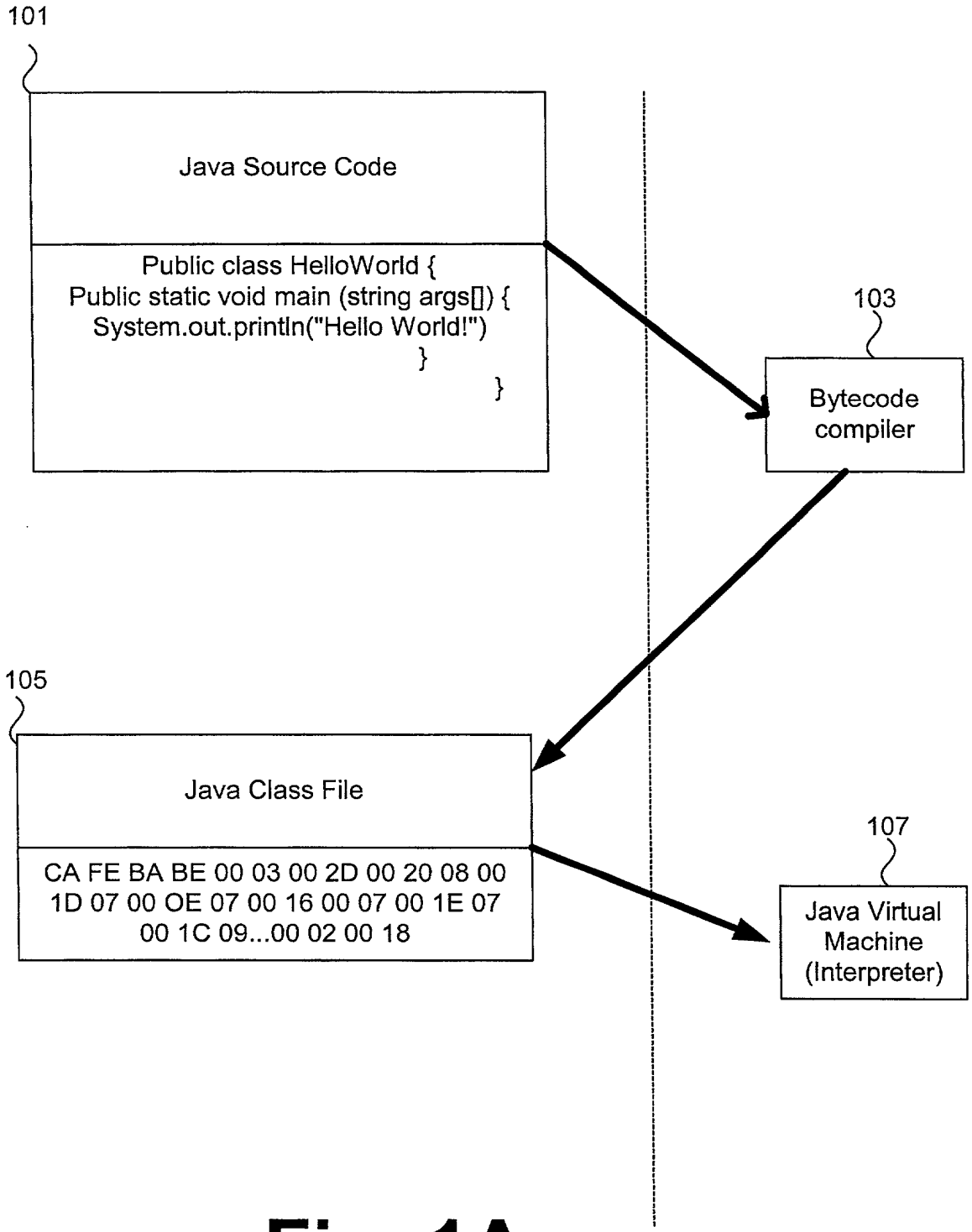
101
Java Source Code
Public class HelloWorld {
Public static void main (string args[]) {
System.out.println("Hello World!")
}
}
103
Bytecode compiler
105
Java Class File
CA FE BA BE 00 03 00 2D 00 20 08 00
1D 07 00 OE 07 00 16 00 07 00 1E 07
00 1C 09...00 02 00 18
107
Java Virtual Machine (Interpreter)

Fig. 1A

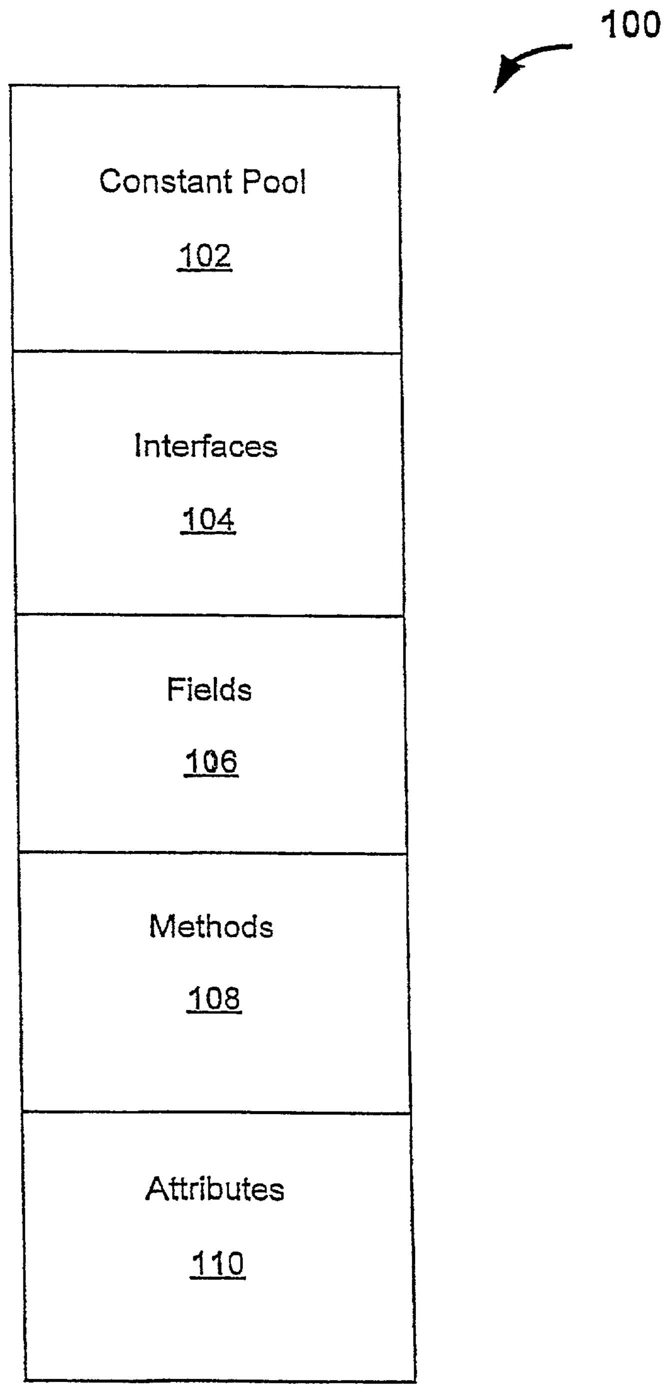
100
Constant Pool
102
Interfaces
104
Fields
106
Methods
108
Attributes
110

Fig. 1B

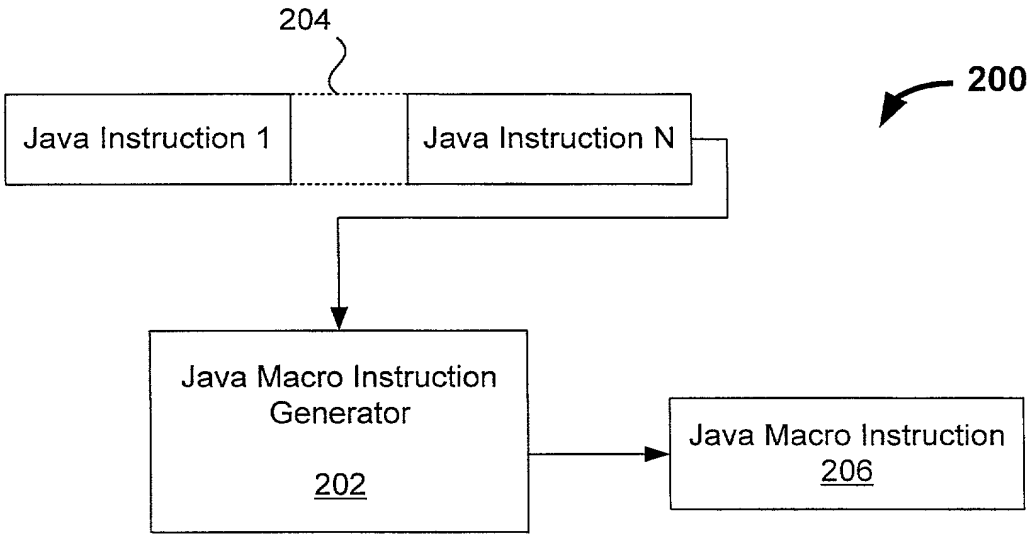
204
200
Java Instruction 1
Java Instruction N
Java Macro Instruction Generator
202
Java Macro Instruction
206

Fig. 2A

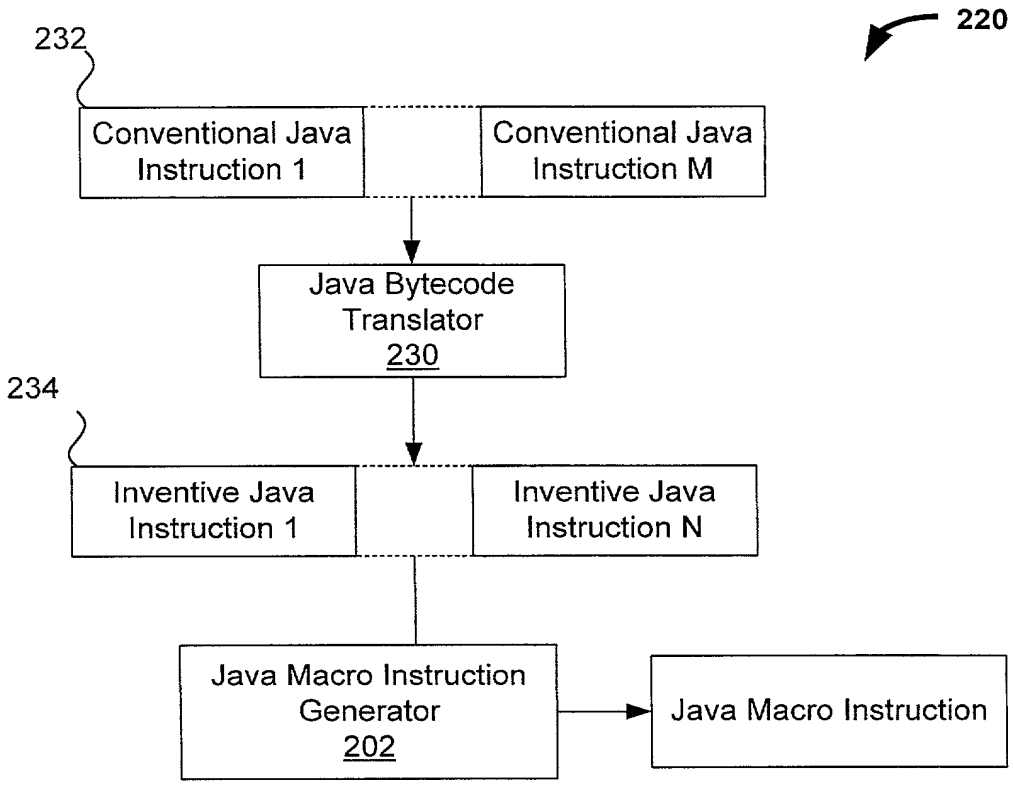
220
232
Conventional Java Instruction 1
Conventional Java Instruction M
Java Bytecode Translator 230
234
Inventive Java Instruction 1
Inventive Java Instruction N
Java Macro Instruction Generator 202
Java Macro Instruction

Fig. 2B

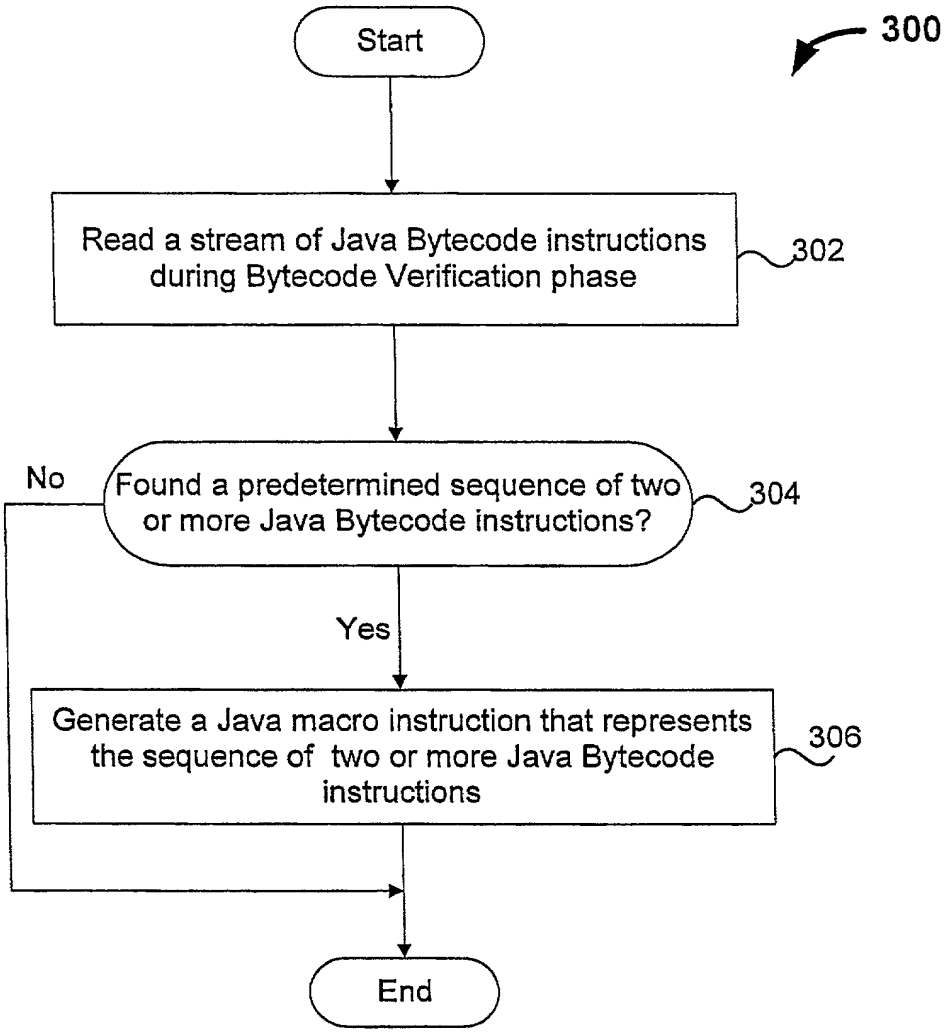
Start
300
Read a stream of Java Bytecode instructions during Bytecode Verification phase
302
No
Found a predetermined sequence of two or more Java Bytecode instructions?
304
Yes
Generate a Java macro instruction that represents the sequence of two or more Java Bytecode instructions
306
End

Fig. 3

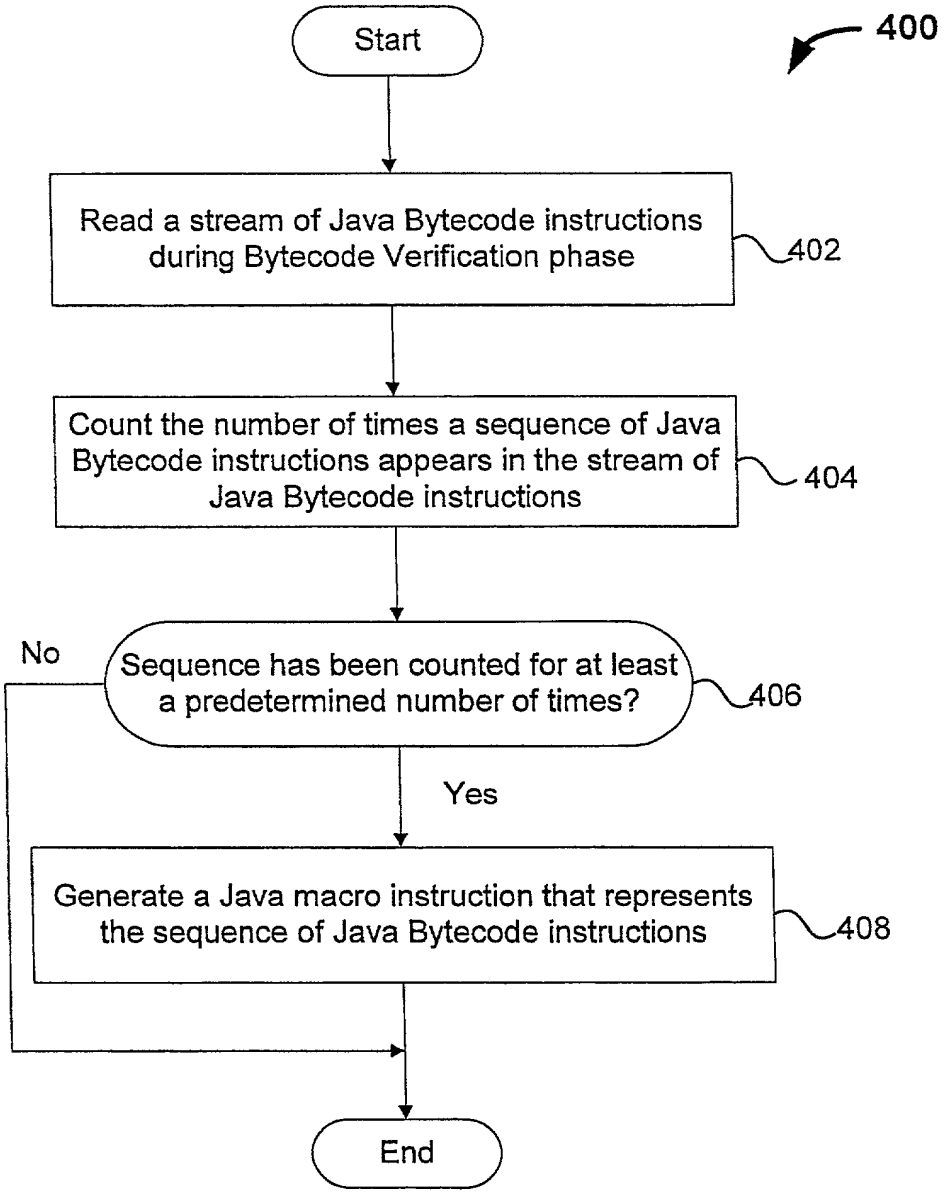
400
Start
Read a stream of Java Bytecode instructions during Bytecode Verification phase
402
Count the number of times a sequence of Java Bytecode instructions appears in the stream of Java Bytecode instructions
404
No
Sequence has been counted for at least a predetermined number of times?
406
Yes
Generate a Java macro instruction that represents the sequence of Java Bytecode instructions
408
End

Fig. 4

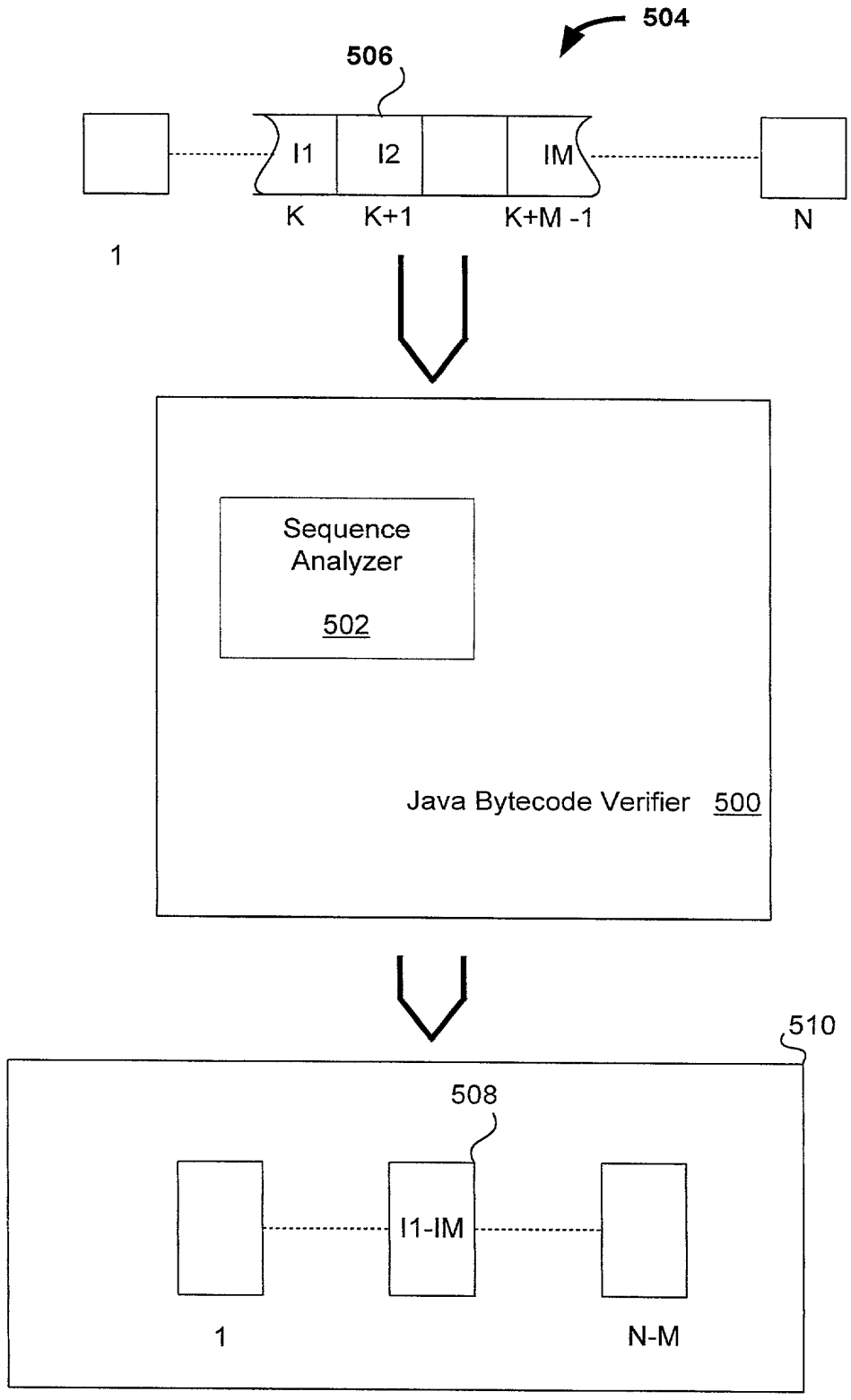
504
506
I1
I2
IM
K
K+1
K+M -1
N
1
Sequence Analyzer
502
Java Bytecode Verifier 500
510
508
I1-IM
1
N-M

Fig. 5

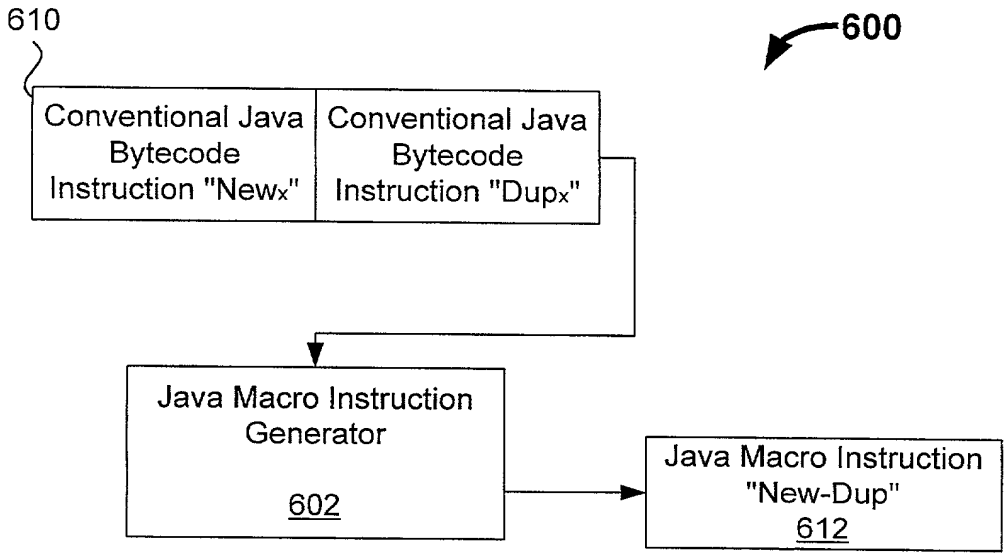
610
600
Conventional Java Bytecode Instruction "Newx"
Conventional Java Bytecode Instruction "Dupx"
Java Macro Instruction Generator
602
Java Macro Instruction "New-Dup"
612

Fig. 6A

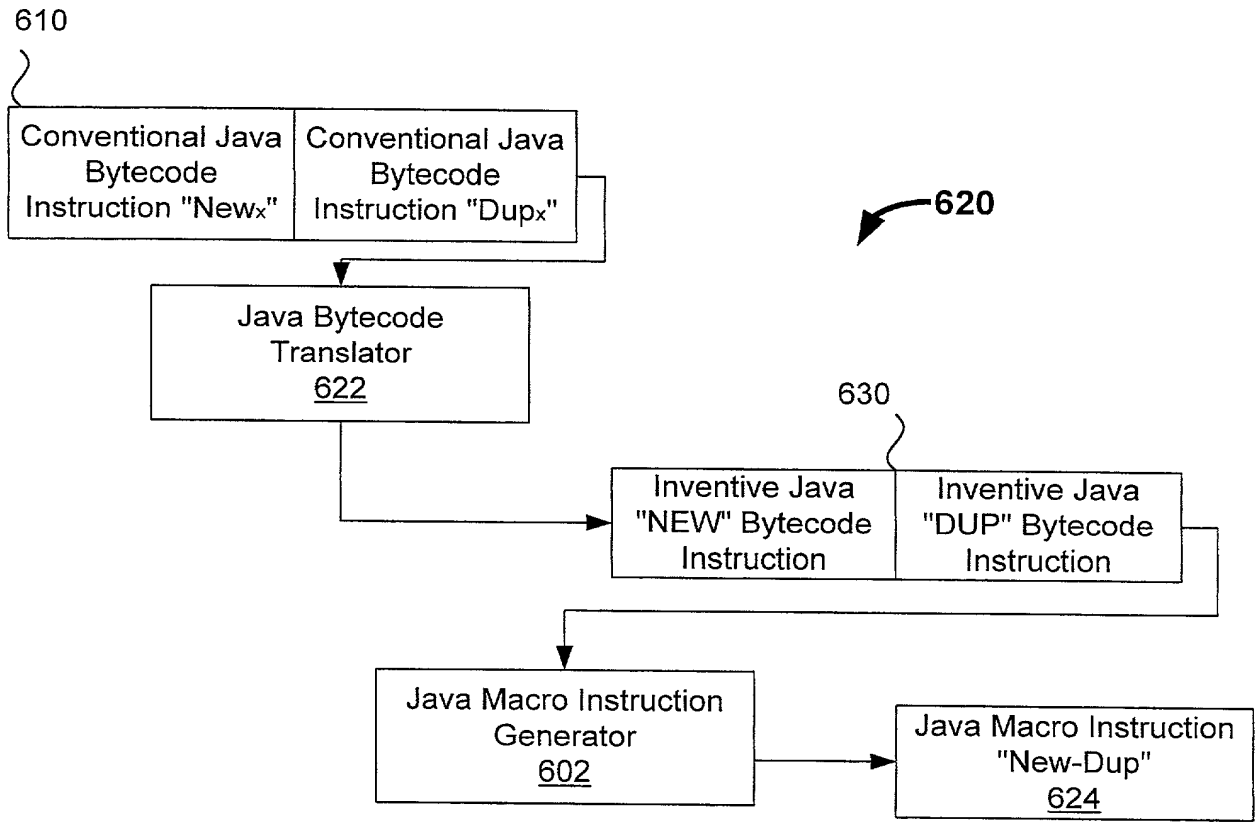
610
620
Conventional Java Bytecode Instruction "Newx"
Conventional Java Bytecode Instruction "Dupx"
Java Bytecode Translator
622
630
Inventive Java "NEW" Bytecode Instruction
Inventive Java "DUP" Bytecode Instruction
Java Macro Instruction Generator
602
Java Macro Instruction "New-Dup"
624

Fig. 6B

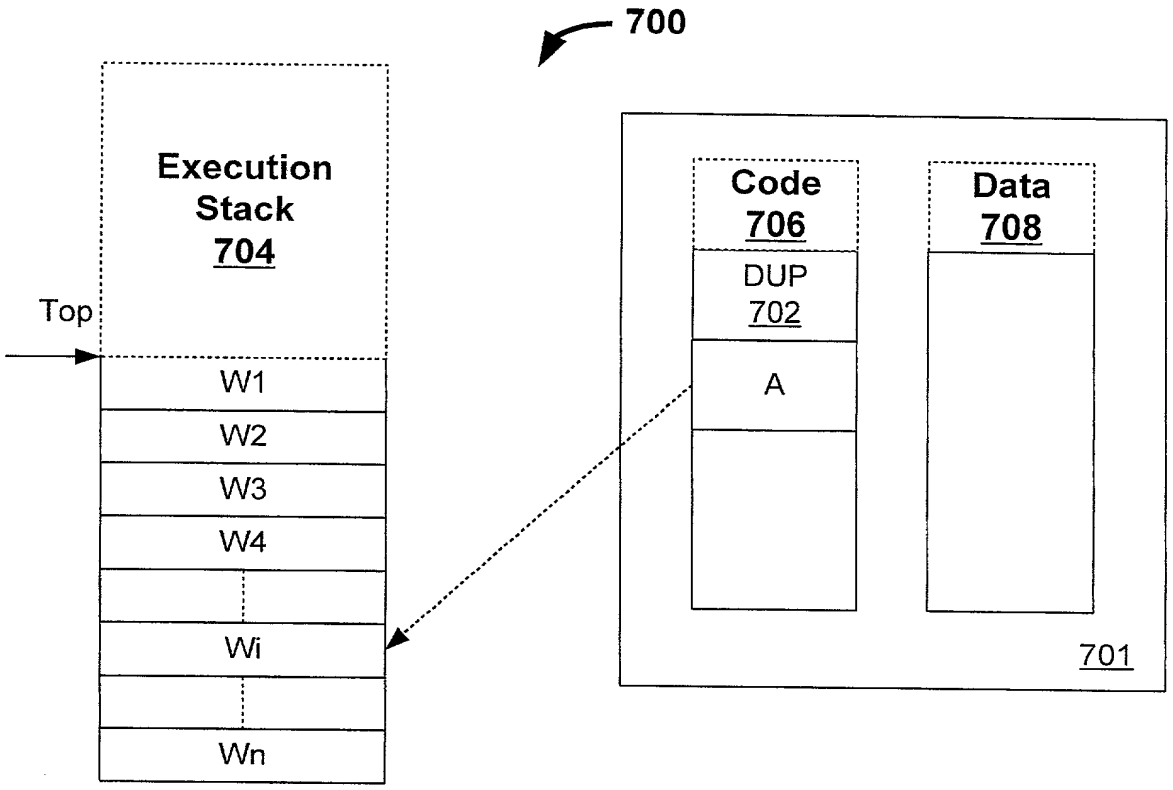

Fig. 7A

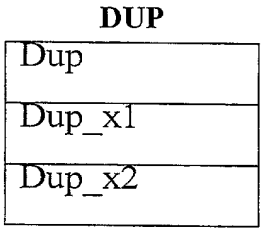

Fig. 7B

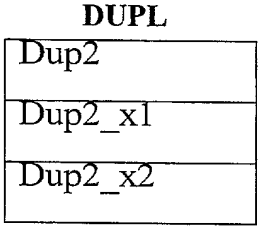

Fig. 7C

Fig. 8

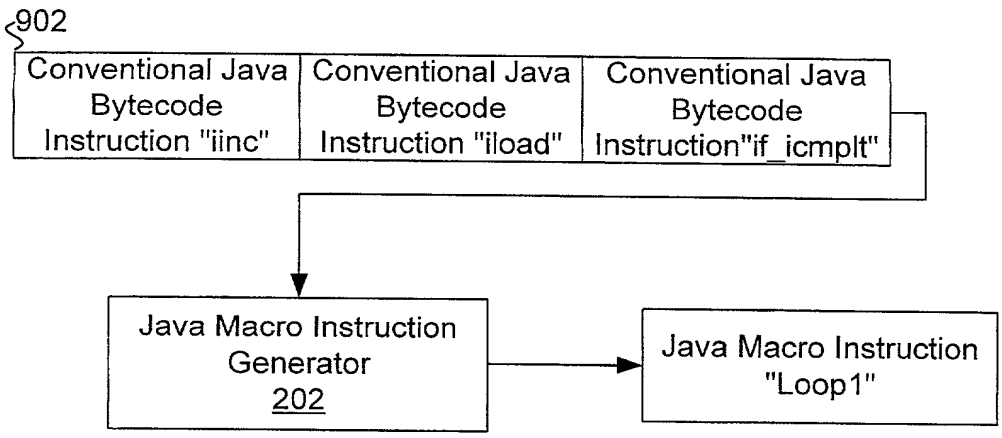
902
Conventional Java Bytecode Instruction "iinc"
Conventional Java Bytecode Instruction "iload"
Conventional Java Bytecode Instruction"if_icmplt"
Java Macro Instruction Generator 202
Java Macro Instruction "Loop1"

Fig. 9A

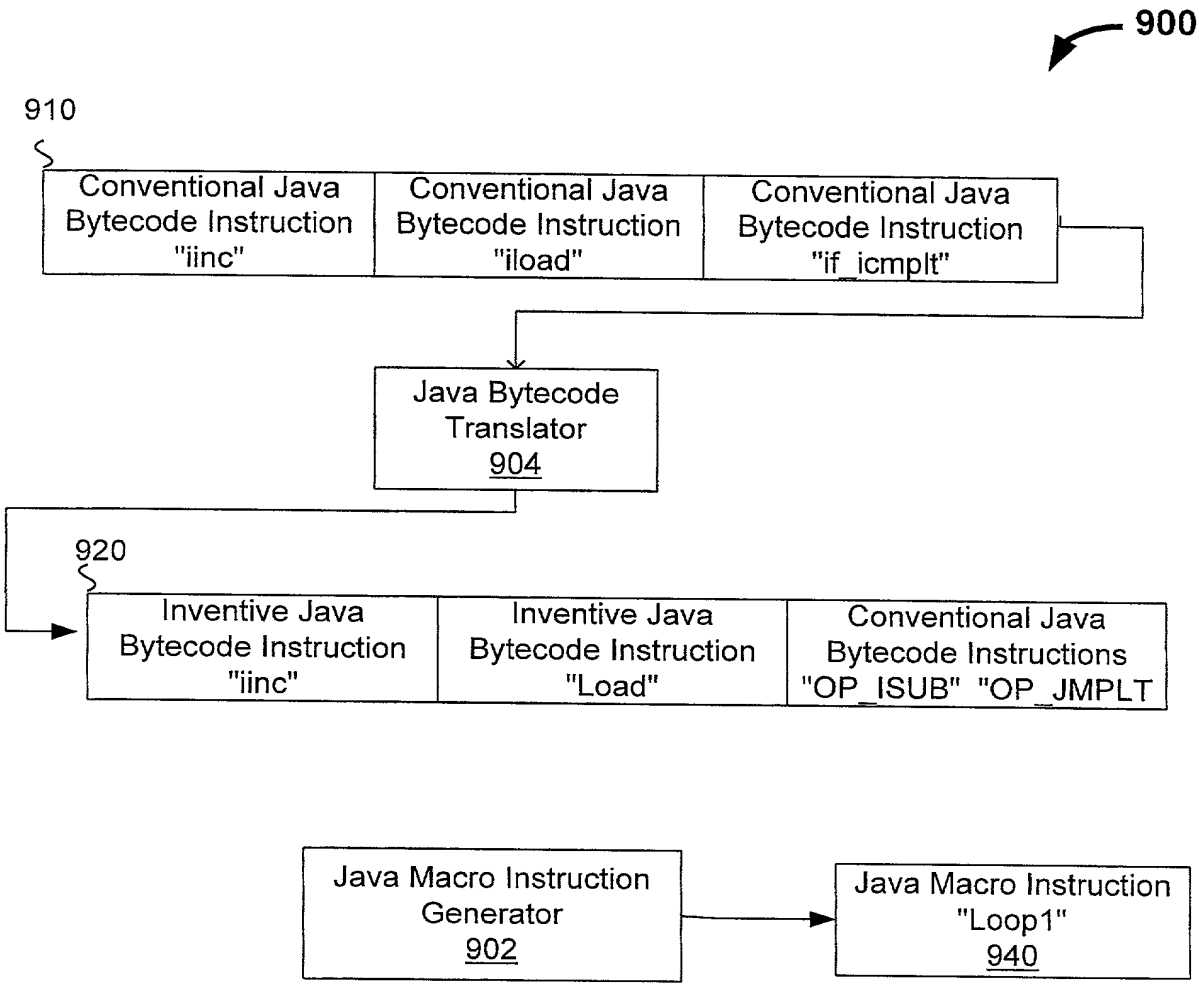
900
910
Conventional Java Bytecode Instruction "iinc"
Conventional Java Bytecode Instruction "iload"
Conventional Java Bytecode Instruction "if_icmplt"
Java Bytecode Translator 904
920
Inventive Java Bytecode Instruction "iinc"
Inventive Java Bytecode Instruction "Load"
Conventional Java Bytecode Instructions "OP_ISUB" "OP_JMPLT
Java Macro Instruction Generator 902
Java Macro Instruction "Loop1" 940

Fig. 9B

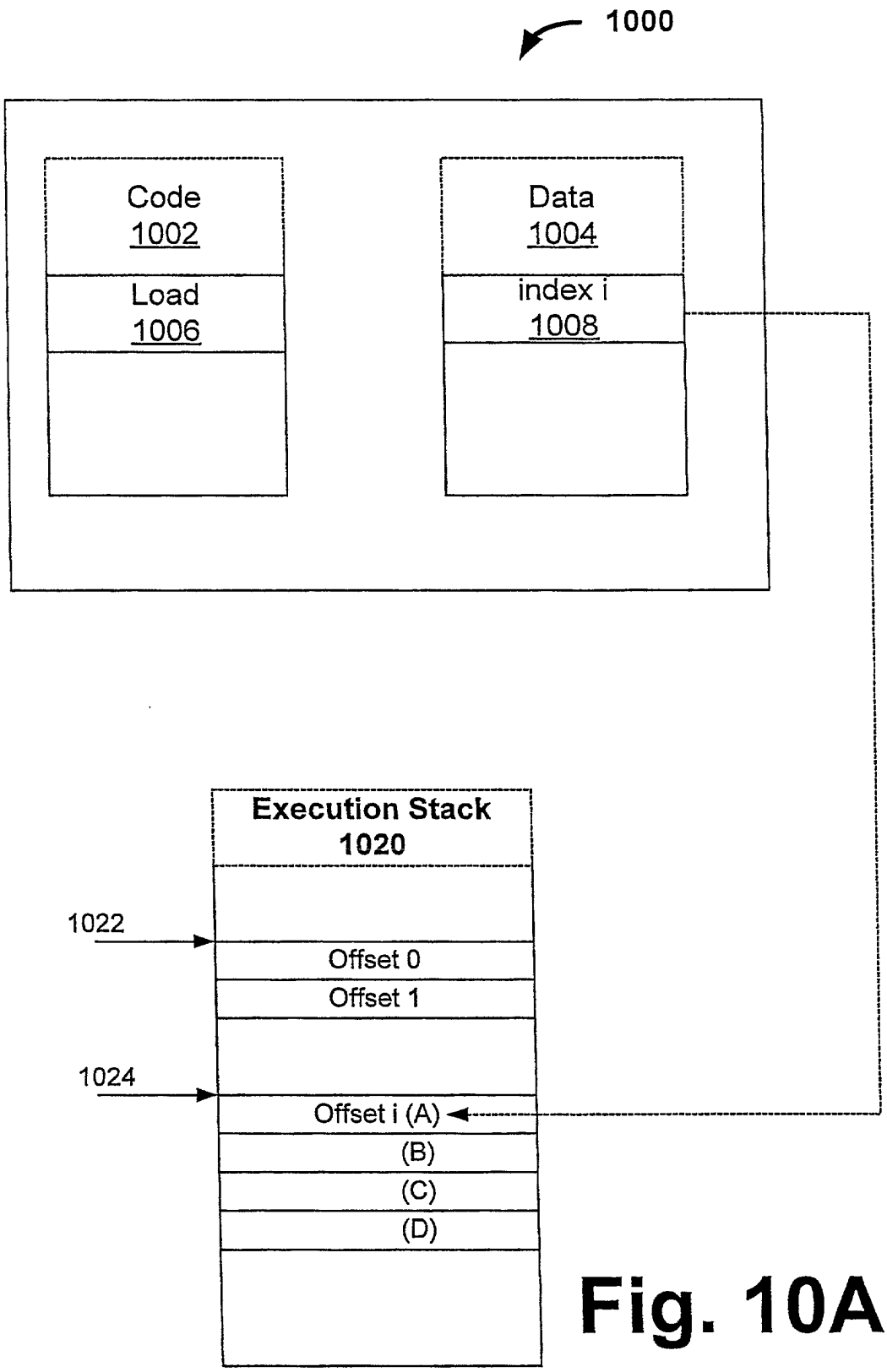
1000
Code
1002
Load
1006
Data
1004
index i
1008
Execution Stack
1020
1022
Offset 0
Offset 1
1024
Offset i (A)
(B)
(C)
(D)

Fig. 10A

LOAD
iload
fload
aload
iload_0
iload_1
iload_2
iload_3
fload_1
fload_2
fload_3
aload_0
aload_1
aload_2
aload_3

LOADL
lload
dload
lload_0
lload_1
lload_2
lload_3
fload_0
dload_0
dload_1
dload_2
dload_3

Fig. 10B

Fig. 10C

lcmp	OP_LSUB, OP_JMPEQ
fcmpl	OP_FSUB, OP_JMPLE
fcmpg	OP_FSUB, OP_JMPGE
dcmpl	OP_DCMP, OP_JMPLE
dcmpg	OP_DCMP, OP_JMPGE

Fig. 11A

if_icmpeq	OP_ISUB, OP_JMPEQ
if_icmpne	OP_ISUB, OP_JMPNE
if_icmplt	OP_ISUB, OP_JMPLT
if_icmpge	OP_ISUB, OP_JMPGE
if_icmpgt	OP_ISUB, OP_JMPGT
if_icmple	OP_ISUB, OP_JMPLE
if_acmpeq	OP_ISUB, OP_JMPEQ
if_acmpne	OP_ISUB, OP_JMPNE

Fig. 11B

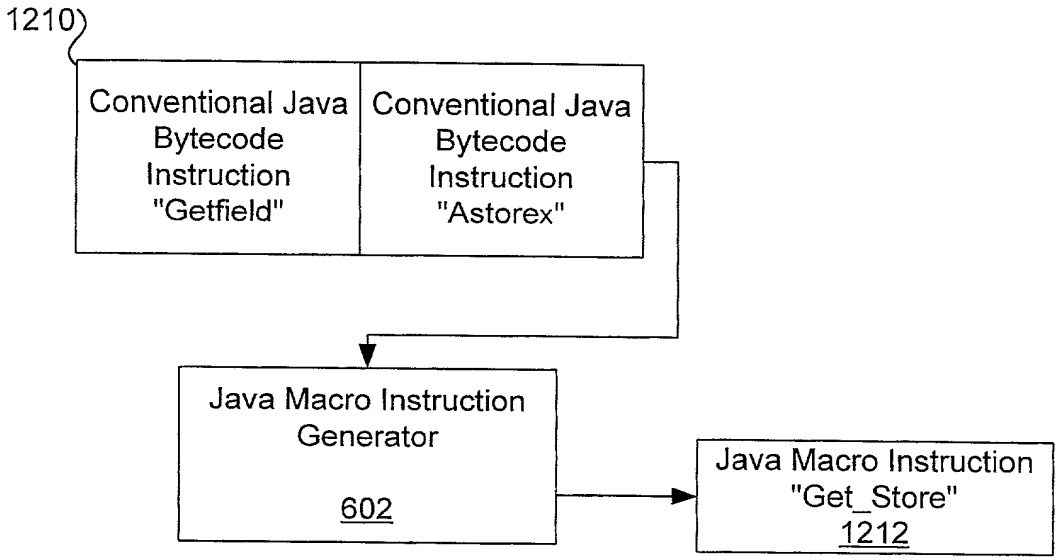

Fig. 12A

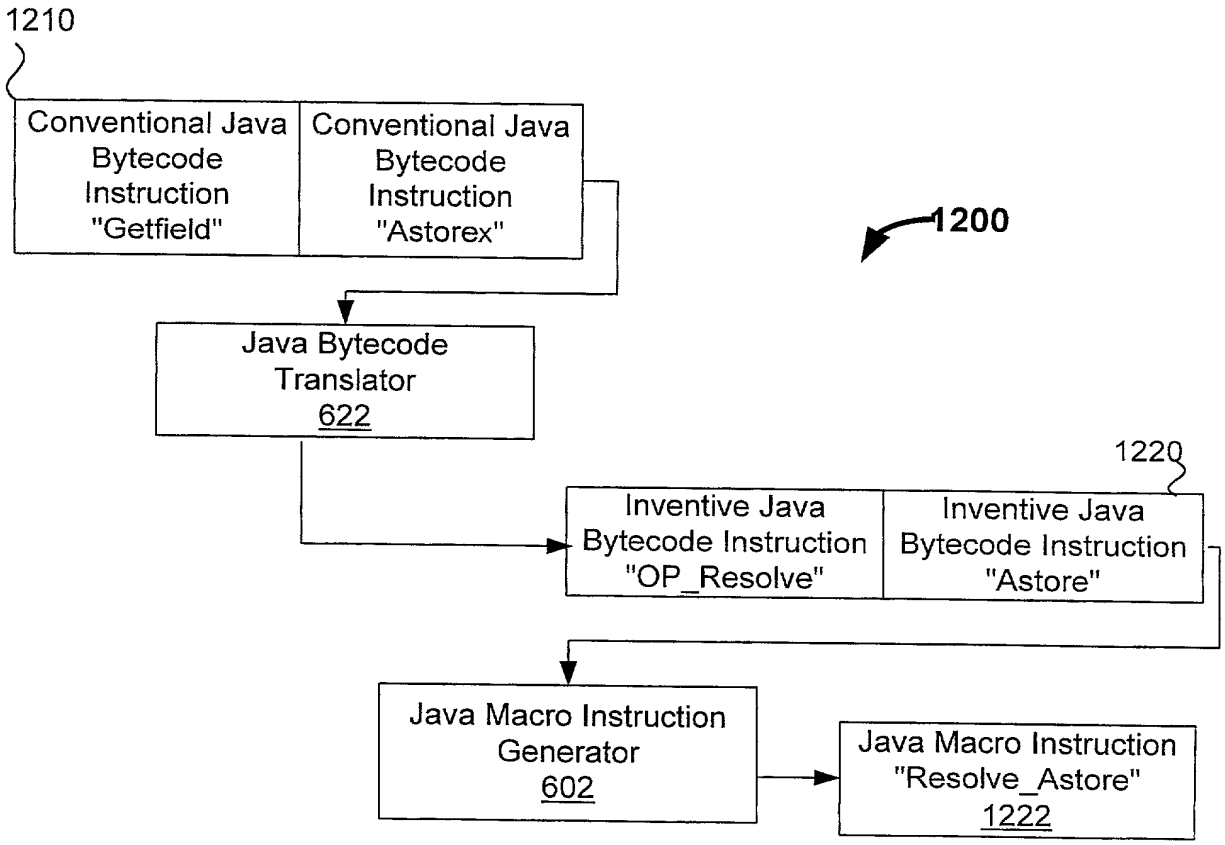

Fig. 12B

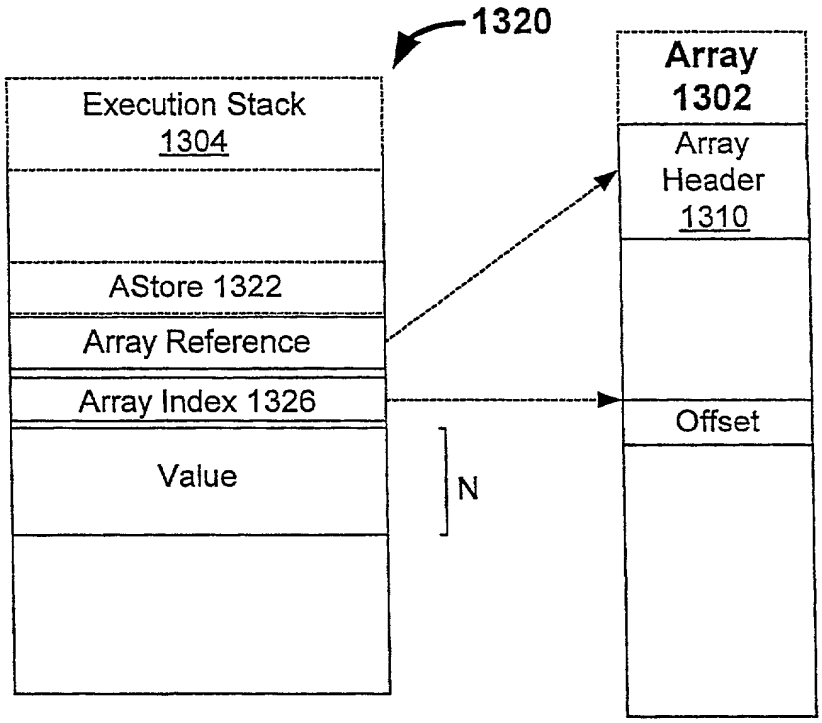

Fig. 13A

ASTORE
iastore
fastore
aastore
bastore
castore
sastore

Fig. 13B

ASTOREL
lastore
dastore

Fig. 13C